Excel Formula and Functions

1. Data Cleaning & Text Manipulation

Core Functions

TRIM

- **Syntax**: `=TRIM(text)`
- **Purpose**: Removes leading/trailing spaces and extra spaces between words.
- **Use Case**: Clean data imported from CSV/APIs (e.g., `=TRIM(A2)`).
- **Pro Tip**: Combine with `CLEAN` to remove non-printable characters: `=TRIM(CLEAN(A2))`.

TEXTJOIN

- **Syntax**: `=TEXTJOIN(delimiter, ignore_empty, text1, [text2], ...)`
- **Purpose**: Joins text with a delimiter, skipping empty cells.
- **Example**: Combine first and last names: `=TEXTJOIN(" ", TRUE, A2, B2)`.
- **Advanced**: Use with `FILTER` to join filtered results: `=TEXTJOIN(", ", TRUE, FILTER(A2:A100, B2:B100 > 100))`.

SUBSTITUTE vs. REPLACE

- **SUBSTITUTE**:
  - **Syntax**: `=SUBSTITUTE(text, old_text, new_text, [instance_num])`
  - *Use Case**: Replace "USD" with "$" in a specific instance: `=SUBSTITUTE(A2, "USD", "$", 2)`.
- **REPLACE**:
  - **Syntax**: `=REPLACE(old_text, start_num, num_chars, new_text)`
  - **Use Case**: Mask credit card numbers: `=REPLACE(A2, 5, 4, "****")`.

TEXTSPLIT (Excel 365)

- **Syntax**: `=TEXTSPLIT(text, col_delimiter, [row_delimiter])`
- **Purpose**: Splits text into rows/columns (dynamic array).
- **Example**: Split comma-separated tags into columns: `=TEXTSPLIT(A2, ",")`.

UPPER, LOWER, PROPER

- **Syntax**:
  - `=UPPER(text)` (Convert to uppercase)
  - `=PROPER(text)` (Capitalize first letters)
- **Use Case**: Standardize names or addresses.

Advanced Techniques

- **Extract Substrings with Regex (Power Query)**:
  Use Power Query's `Text.Select` or `Text.Remove` for regex-like cleaning.
- **Remove Duplicates with `UNIQUE`**:
  `=UNIQUE(A2:C100)` to generate a deduplicated list.
- **Validate Data with `LEN` and `IF`**:
  `=IF(LEN(A2)=10, "Valid", "Invalid")` for 10-digit IDs.

---

2. Lookup & Reference Functions

Core Functions

XLOOKUP (Excel 365)

- **Syntax**:
  `=XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])`
- **Features**:
  - Searches in any direction (replaces VLOOKUP/HLOOKUP).
  - Supports approximate/exact matches and wildcards (`*?`).
- **Example**:
  `=XLOOKUP("Widget", Products, Prices, "Not Found", 0, -1)` (Search from bottom to top).

INDEX-MATCH-MATCH (2-Way Lookup)

- **Syntax**:
  `=INDEX(return_range, MATCH(row_value, row_range, 0), MATCH(col_value, col_range, 0))`
- **Use Case**: Fetch sales for "ProductA" in "Q3":
  `=INDEX(SalesData, MATCH("ProductA", Products, 0), MATCH("Q3", Quarters, 0))`.

INDIRECT for Dynamic References

- **Syntax**: `=INDIRECT(ref_text)`
- **Use Case**: Reference sheets dynamically:
  `=SUM(INDIRECT("'" & A2 & "'!B2:B10"))` (Sum data from a sheet named in cell A2).
- **Warning**: Volatile function—use sparingly in large datasets.

Advanced Techniques

- **Multi-Condition Lookup with `FILTER`**:
  `=FILTER(SalesData, (Regions="East")*(Sales>1000))`.
- **Dynamic Dropdowns with `UNIQUE` and `SORT`**:
  Data Validation List Source:
  `=SORT(UNIQUE(A2:A100))`.

---

3. Logical Functions

Core Functions

IFS for Multiple Conditions

- **Syntax**: `=IFS(condition1, value1, condition2, value2, ...)`
- **Example**: Categorize sales tiers:
  `=IFS(B2>1000, "Platinum", B2>500, "Gold", TRUE, "Silver")`.

SWITCH for Value Mapping

- **Syntax**: `=SWITCH(expression, value1, result1, ..., [default])`
- **Example**: Convert region codes:
  `=SWITCH(A2, "NE", "Northeast", "SW", "Southwest", "Other")`.

Boolean Logic with `AND` / `OR`

- **Syntax**:
  `=IF(AND(B2>100, C2<50), "Approve", "Review")`
- **Pro Tip**: Use `*` for AND and `+` for OR in array formulas:
  `=SUM((Region="East")*(Sales>1000))`.

Advanced Techniques

- **Nested `IF` with `VLOOKUP`**:
  Replace complex nested IFs with a lookup table.
- **Error Handling with `IFERROR` / `IFNA`**:
  `=IFERROR(VLOOKUP(A2, Data, 2, 0), "Not Found")`.

---

4. Aggregation & Statistical Analysis

Core Functions

SUMIFS/COUNTIFS

- **Syntax**:
  `=SUMIFS(sum_range, criteria_range1, criteria1, ...)`
- **Example**: Sum 2023 sales for Product A in the East region:
  `=SUMIFS(Sales, Products, "A", Regions, "East", Years, 2023)`.

SUBTOTAL for Filtered Data

- **Syntax**: `=SUBTOTAL(function_num, range)`
- **Key Function Numbers**:
  - `9` (SUM), `1` (AVERAGE), `2` (COUNT), `3` (COUNTA)
- **Use Case**: Calculate totals ignoring hidden rows:
  `=SUBTOTAL(9, B2:B100)`.

CORREL for Correlation Analysis

- **Syntax**: `=CORREL(array1, array2)`
- **Pro Tip**: Use `=CORREL(Sales, Advertising)` to measure ROI impact.

Advanced Techniques

- **Weighted Average with `SUMPRODUCT`**:
  `=SUMPRODUCT(Weights, Scores)/SUM(Weights)`.
- **Percentile Analysis**:
  `=PERCENTILE.INC(SalesData, 0.9)` (90th percentile).
- **Conditional Median with `AGGREGATE`**:
  `=AGGREGATE(12, 5, Sales, Regions="East")` (Median for East region).

---

5. Date & Time Analysis

Core Functions

EOMONTH for Month-End Dates

- **Syntax**: `=EOMONTH(start_date, months)`
- **Example**: `=EOMONTH(TODAY(), -1)` returns last month's end date.

NETWORKDAYS.INTL for Custom Calendars

- **Syntax**:
  `=NETWORKDAYS.INTL(start_date, end_date, [weekend], [holidays])`
- **Example**: Calculate workdays with Saturday/Sunday weekends:
  `=NETWORKDAYS.INTL(A2, B2, 1)`.

DATEDIF for Date Differences

- **Syntax**: `=DATEDIF(start_date, end_date, unit)`
- **Units**: "Y" (years), "M" (months), "D" (days).
- **Use Case**: Calculate employee tenure:
  `=DATEDIF(A2, TODAY(), "Y") & " Years"`.

Advanced Techniques

- **Dynamic Date Ranges with `SEQUENCE`**:
  `=TEXT(SEQUENCE(30, , TODAY()), "mmm dd")` generates the next 30 days.
- **Time-Series Forecasting with `FORECAST.LINEAR`**:
  `=FORECAST.LINEAR(A2, HistoricalSales, HistoricalDates)`.

---

6. Dynamic Arrays (Excel 365)

Core Functions

FILTER with Multiple Criteria

- **Syntax**: `=FILTER(array, (criteria1)*(criteria2), "No Data")`
- **Example**: Filter high-priority East region sales:
  `=FILTER(SalesData, (Regions="East")*(Priority="High"))`.

SORT/SORTBY

- **Syntax**:
  - `=SORT(range, [sort_index], [sort_order])`
  - `=SORTBY(range, by_array1, [sort_order1], ...)`
- **Example**: Sort sales descending and then by region:
  `=SORTBY(SalesData, Sales, -1, Regions, 1)`.

SEQUENCE for Data Generation

- **Syntax**: `=SEQUENCE(rows, [columns], [start], [step])`
- **Use Case**: Create a date sequence:
  `=TODAY() + SEQUENCE(30)` (Next 30 days).

Advanced Techniques

- **Spill Ranges**: Chain dynamic arrays (e.g., `=SORT(UNIQUE(...))`).
- **Combining `FILTER` and `XLOOKUP`**:
  `=XLOOKUP("Widget", Products, FILTER(Sales, Regions="East"))`.

---

7. Error Handling & Optimization

Core Functions

IFERROR vs. IFNA

- **IFERROR**: Catches all errors (`#N/A`, `#VALUE!`, etc.):
  `=IFERROR(VLOOKUP(...), "Not Found")`.
- **IFNA**: Catches only `#N/A` errors (safer for debugging):
  `=IFNA(VLOOKUP(...), "Not Found")`.

LET for Readability (Excel 365)

- **Syntax**: `=LET(name1, value1, name2, value2, ..., calculation)`
- **Example**: Simplify complex formulas:

```
=LET(
  Sales, B2:B100,
  Total, SUM(Sales),
  Total/COUNT(Sales)
)
```

Pro Tips

- **Avoid Volatile Functions**: Minimize `TODAY()`, `RAND()`, `OFFSET` in large datasets.
- **Use Named Ranges**: `=SUM(Sales)` instead of `=SUM(B2:B100)`.
- **Optimize Formulas**: Replace `VLOOKUP` with `XLOOKUP` for speed.
- **Power Query for Big Data**: Clean/transform millions of rows efficiently.

---

8. Advanced Analytics & Automation

Power Query (Get & Transform)

- **Key Actions**:
  - Merge tables (like SQL JOINs).
  - Unpivot data for analysis.
  - Group/aggregate with custom logic.

PivotTables

- **Pro Tips**:
  - Use "Show Values As" for % of total or YoY growth.
  - Add calculated fields: `=Sales - Cost`.

Macros & VBA

- **Example**: Automate report generation with:

```
Sub RefreshReport()
  ThisWorkbook.RefreshAll
  Sheets("Dashboard").PivotTables("SalesPivot").RefreshTable
End Sub
```

Ultimate Example Formula

```
=LET(
  FilteredData, FILTER(SalesData, (Regions="East")*(Sales>1000)),
  SortedData, SORTBY(FilteredData, INDEX(FilteredData, , 3), -1),
  IFERROR(INDEX(SortedData, , {1,3}), "No Data")
)
```

Explanation:

1. Filters East region sales > $1000.

2. Sorts by the 3rd column (Sales) descending.
3. Returns columns 1 (Product) and 3 (Sales), handling errors gracefully.

Final Tips for Data Analysts

1. **Master Power Query**: It's faster than formulas for ETL.
2. **Use Excel Tables**: Automatic structured references and dynamic ranges.
3. **Learn Array Formulas**: Use `CTRL+SHIFT+ENTER` in older Excel versions.
4. **Leverage Conditional Formatting**: Highlight outliers with `=AND(A2>AVERAGE(A:A), A2>1000)`.

With these tools, you'll handle everything from basic data cleaning to advanced predictive analytics in Excel! 🚀📈